

Serial No. 09/456,894

REMARKS

The Applicant and the undersigned thank Examiner El-Chanti for his consideration given during the telephonic interview of May 18, 2006 and his careful review of this application.

Upon entry of this amendment, Claims 1-2, 4-12, 14-26, and 28-29 remain pending in this application. Claims 1-2, 4-12, 14-26 and 28-29 are being prosecuted while Claims 30-58 have been cancelled. The independent claims are Claims 1, 11, 18, 23, 26, and 28. Consideration of the present application is respectfully requested in light of the above amendments to the application and in view of the following remarks.

Summary of Telephonic Interview of May 18, 2006 - Pursuant to M.P.E.P. § 713.04 and 37 C.F.R. § 1.133(b)

The Applicant and the undersigned appreciate the time and consideration given by Examiner El Chanti during the telephonic interview of May 18, 2006. During this telephonic interview, a proposed claim amendment was presented and discussed. Applicant's representative emphasized that the Examiner's primary prior art reference of U.S. Patent No. 6,009,274 issued in the name of Fletcher et al. (hereinafter the "Fletcher reference") does not provide any teaching of a template jump for forwarding call instructions from the network driver interface to a template. Applicants' representative further explained that the Fletcher reference does not patch a network driver software interface while the software interface is running because the Fletcher reference expressly teaches that all operations are stopped before any new code is executed. See column 13, lines 50-52 of the Fletcher reference which explains how services are stopped while updates are provided.

Examiner El-Chanti already acknowledged that the Fletcher reference did not provide any teaching of template jumps. To make up for this deficiency, Examiner El-Chanti relied upon U.S. Patent No. 5,781,776 issued in the name of Johnston et al. (hereinafter the "Johnston reference") to address template jumps. It was explained to Examiner El-Chanti that the Johnston reference was in the field of industrial controllers and did not relate in any way to modifying or

Serial No. 09/456,894

patching network software interfaces that provide communication between one or more media access control units and one or more protocol drivers in a computer system, as recited in amended independent Claim 1.

Upon further review of the proposed amendment, Examiner El-Chanti informed Applicants' representative that he believed that some of the new language of the proposed claim amendment was not clear. Examiner El-Chanti stated that the element for receiving the call instructions from the template jump as described in the faxed proposed amendment was not clear. Examiner El-Chanti also requested that the Applicant clarify how access to the patched first section of the computer code is re-enabled. Specifically, Examiner El-Chanti requested that the Applicant clarify the phrase, "...by replacing the blocking computer code with code that allows..." that was recited in Claim 1 of the faxed proposed amendment.

The Applicant's representative advises that the two very helpful suggestions made by Examiner El-Chanti during the telephonic interview and discussed above have been adopted in this paper. Specifically, the element receiving the forwarded call instructions, as recited in the amended claims, is now identified as the template. The template has been further defined in the independent claims. Support for this aspect of the claimed invention can be found in the original description on page 20, line 23 through page 21, line 19; See also Figure 12. Also, the "replacing" phrase noted above has been amended to state: --by replacing the blocking computer code with the patched first section of code--.

At the end of the telephonic interview, Examiner El Chanti explained that he would investigate these new claim elements further and that he would likely conduct an update search upon the filing of an RCE by the Applicant.

Consideration and approval of this interview summary by Examiner El Chanti are respectfully requested. The Examiner is requested to initial this summary if he approves of it pursuant to M.P.E.P. § 713.04.

Restriction Requirement

The Examiner imposed a restriction requirement on January 13, 2005. The Applicant elected with traverse Claims 1-26 and 28-29, while Claims 30, 33-40, 42-53, and 55-57 would

Serial No. 09/456,894

stand withdrawn as non-elected claims. The Examiner has made the restriction requirement final. The Applicant has cancelled the withdrawn claims without prejudice or disclaimer in order to advance prosecution of this patent application. The Applicant has preserved the right to pursue these cancelled patent claims in a later filed divisional patent application.

Response to Examiner's Comments on Information Disclosure Statement (IDS) submitted on October 11, 2005

The Examiner advises the Applicant that the IDS submitted on October 11, 2005 fails to comply with 37 C.F.R. § 1.98(a) because it does not include a concise explanation of relevance as it is presently understood by the individual designated in 37 C.F.R. § 1.56(c) most knowledgeable about the content of the information *of each patent listed that is not in the English language* (emphasis supplied by Applicant). The Applicant submits that there are no patents or publications listed in the IDS of October 11, 2005 that are not in English.

However, even if Applicant had submitted a publication or patent that is not in the English language, the Applicant submits that the Examiner must consider the other publications that are in compliance with the rule and that the Examiner should merely cross through those entries that are not in compliance with 37 C.F.R. § 1.98(a). See M.P.E.P. § 609.01, subsection (B), page 600-143 (Rev. 3, Aug. 2005).

The Applicant respectfully submits that one or more entries of an IDS that may NOT be in compliance do not affect the status of other entries that may be IN compliance with the Rule. Therefore, appropriate consideration of those entries that are in compliance where the Examiner initials the PTO-1449 adjacent to compliant entries is respectfully requested.

Rejections under 35 U.S.C. § 103(a)

The Examiner rejected Claims 1, 2, 4-12, 14-26, and 28-29 under 35 U.S.C. § 103(a) as being unpatentable over the Fletcher reference in view of the Johnston reference. The Applicant respectfully offers remarks to traverse these rejections. The Applicant will address each independent claim separately as the Applicant believes that each independent claim is separately patentable over the prior art of record.

Serial No. 09/456,894

Independent Claim 1

The rejection of Claim 1 is respectfully traversed. It is respectfully submitted that the Fletcher and Johnston references, alone or in combination, fail to describe, teach, or suggest the combination of: (1) disabling access to at least a first section of computer code in (2) a network driver software interface (3) that is being executed by the processor (4) by overwriting computer code that is executed before the first section of computer code (5) with blocking computer code, wherein (6) the network driver software interface provides for communication between one or more media access control units and one or more protocol drivers in a computer system according to a set of bindings; (7) executing the blocking computer code with the processor; (8) patching the first section of computer code (8) while the blocking computer code of the network driver software interface is being executed by the processor (9) and without stopping complete operation of the network driver software interface, (10) the patching of the first section of code comprising inserting a template jump (11) for forwarding call instructions from the network driver software interface to a template in a rerouting driver, (12) the template comprising new computer code for controlling communications, (13) the template jump and template allowing the rerouting driver to control communication between one or more media access control units and one or more protocol drivers in the computer system; and (14) re-enabling access to the patched first section of computer code by (15) replacing the blocking computer code with the patched first section of computer code, as recited in amended independent Claim 1.

The Fletcher Reference

The Fletcher reference describes how Windows 95 operating system registries can be updated to point to new file components (updates). Specifically, the Fletcher reference explains that when an Windows 95 operating system is rebooted, the modified registry information can be used by the operating system components.

While the Examiner may argue that the Fletcher reference is in the Applicant's field of endeavor, one of ordinary skill in the art recognizes that is the only extent of relevance to the

Serial No. 09/456,894

claimed subject matter: the Fletcher reference merely teaches updates. What is important to note about the Fletcher reference is that system components are stopped completely, unloaded, or re-booted in order to run or execute any new updates.

For example, the Fletcher reference explains in column 13, lines 40-55, (a) how a network interface card (NIC) driver is temporarily unloaded; (b) how protocols are unbound from the NIC driver; and (c) how system service is stopped so that the NIC driver can be updated:

“According to one embodiment of the invention, system configuration managers and service managers are used to dynamically update NIC drivers and services while the network is operating (i.e., without having to reboot). The system configuration manager in Windows 95, for example can be used to temporarily unload the NIC driver and unbind the protocols currently bound to it. The configuration manager then reloads the NIC driver, which is the new version, and notifies the different protocols installed on the system to rebind to the NIC driver, all without physically rebooting the system. Using the system service manager of Windows NT, for example, a service can be temporarily stopped and updated without physically rebooting the system. A service is any node that resides in the kernel of the system software that is hidden to the user, and which always launches and runs whether or not a specific user is logged into the system.”

Opposite to this stopping of system components of the Fletcher reference, the network driver software interface of the invention is not unloaded or stopped. Instead, a first section of computer code is patched while the blocking computer code of the network driver software interface is being executed by the processor and without stopping complete operation of the network driver software interface, as recited in amended independent Claim 1. As a further distinguishing features, the Fletcher reference also does not provide any teaching of (a) template jumps for forwarding call instructions from the network driver software interface to a template in a rerouting driver; (b) the template comprising new computer code for controlling communications; and (c) the template jump and template allowing the rerouting driver to control communication between one or more media access control units and one or more protocol drivers in the computer system, as recited in amended independent Claim 1. The Examiner has

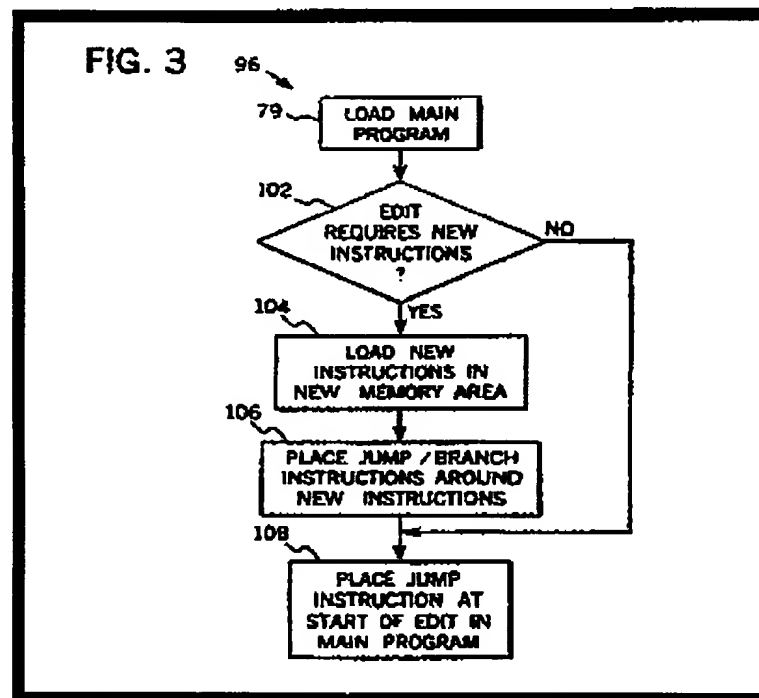
Serial No. 09/456,894

already admitted in the last Office Action of December 16, 2005 that the Fletcher reference does not provide any teaching of template jumps.

The Johnston Reference

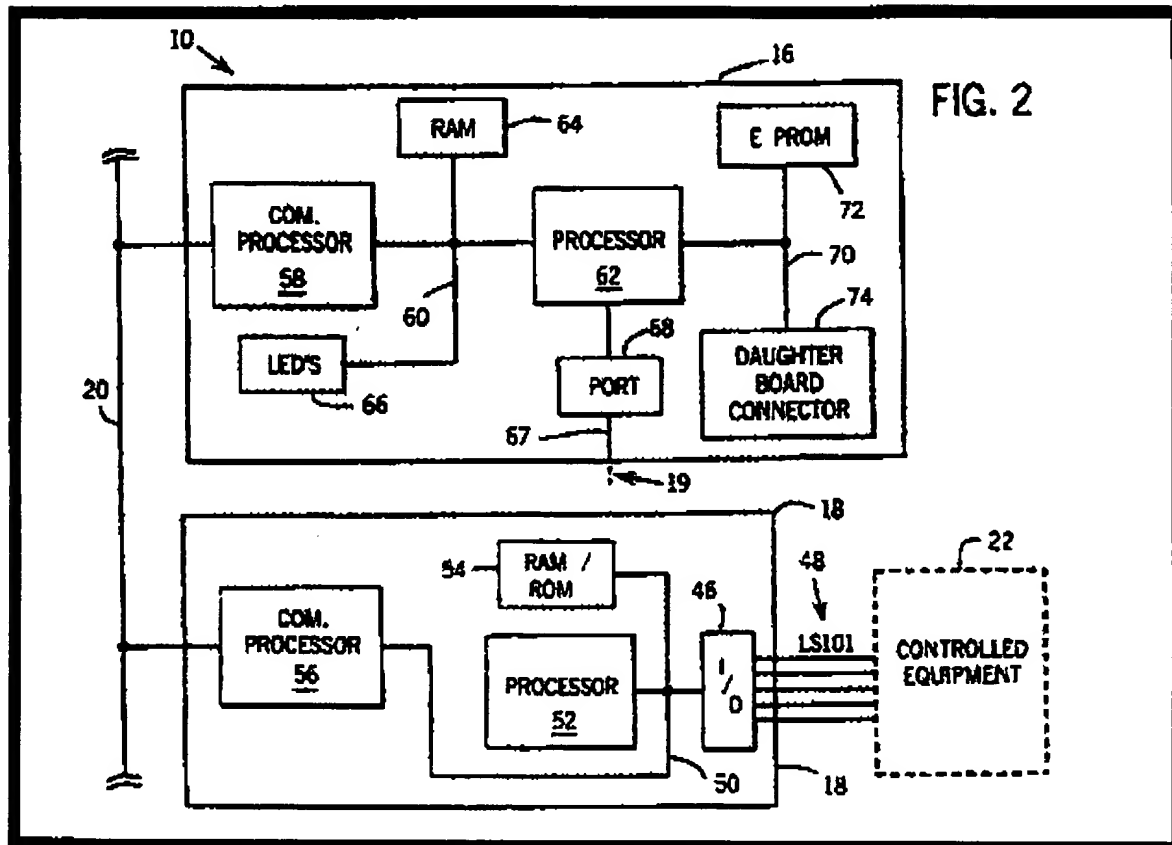
The Johnston reference describes a method for editing a real-time control program as it controls equipment. Editing of the control program occurs in a second area of memory and is integrated into the pre-existing program by a conditional jump instructions that are concatenated to the edited material. See Johnston reference, Abstract.

Specifically, the Johnston reference illustrates in Figure 3 (listed below) a flow chart where editing operations including deletions, insertions, and replacements are undertaken in a separate area of memory (step 104) and then activated with the insertion of a single jump instruction in the pre-existing program (step 106):



Serial No. 09/456,894

The Johnston reference further explains that the editing program 96 of Figure 3 may reside within an EPROM 72 as illustrated in Figure 2 (listed below). The editing program 96 may be executed concurrently with the control program by processor 62 according to time sharing techniques. See Johnston reference, column 8, lines 14-25.



One of ordinary skill in the art recognizes that the Johnston reference generally describes industrial controllers and does not relate in any way to modifying or patching network software interface software that provide communication between one or more media access control units and one or more protocol drivers in a computer system. And because the Johnston reference does not address network software interfaces, the Johnston reference fails to provide any teaching of a template comprising new computer code for controlling communications, or a template jump and the template allowing the rerouting driver to control communication between one or more media access control units and one or more protocol drivers in the computer system.

Serial No. 09/456,894

Summary for Independent Claim 1

Therefore, the Fletcher and Johnston references, alone or in combination, simply do not teach all of the computer code level elements and operations for a single network driver software interface as recited in amended independent Claim 1. In light of the differences between Claim 1 and the Fletcher and Johnston references noted above, one of ordinary skill in the art recognizes that these two references, alone or in combination, cannot anticipate or render obvious the recitations as set forth in amended independent Claim 1. Accordingly, reconsideration and withdrawal of this rejection are respectfully requested.

Independent Claim 11

The rejection of Claim 11 is respectfully traversed. It is respectfully submitted that the Fletcher and Johnston references, alone or in combination, fail to describe, teach, or suggest the combination of: (1) transmitting from a remote host to a first target computer on a network an installation application and a rerouting driver; (2) transmitting from the remote host to the first target computer a command to cause the first target computer to execute the installation application; (3) the first target computer, responsive to receipt of the command, executing the installation application, (4) wherein the first target computer includes a network driver software interface that provides for communication between one or more media access control units and one or more protocol drivers according to a set of bindings; and (5) the first target computer, responsive to executing the installation application, causing the modification of the network driver software interface to insert the rerouting driver into the one or more communication paths provided by (6) the set of bindings (7) while the network driver software interface is being executed by the first target computer and (8) without restarting the first target computer, (9) the first target computer comprising a multiprocessor system, wherein the insert of the rerouting driver, further comprises: (10) the installation application disabling access to a least a first section of code in the network driver software interface (11) by overwriting code prior to the first section with blocking code and (12) without stopping complete operation of the network driver software interface; (13) the installation application patching the first section of code (14) while

Serial No. 09/456,894

the blocking code is being executed by the processor, (15) the patching comprising inserting a template jump (16) for forwarding call instructions from the network driver software interface to a template in the rerouting driver, (17) the template comprising new computer code for controlling communications, (18) the template jump and template allowing the rerouting driver to control communication between the one or more media access control units and the one or more protocol drivers, as recited in amended independent Claim 11.

Similar to what is discussed above with respect to independent Claim 1, the Fletcher and Johnston references do not provide any teaching of (a) patching a network driver software interface, wherein the patching comprises (b) inserting a template jump (c) for forwarding call instructions from the network driver software interface to a template in the rerouting driver, (d) the template comprising new computer code for controlling communications, (e) the template jump and template allowing the rerouting driver to control communication between the one or more media access control units and the one or more protocol drivers, as recited in amended independent Claim 11.

In light of the differences between the claims and the Fletcher and Johnston references mentioned above, one of ordinary skill in the art recognizes that these two references alone or in combination cannot anticipate or render obvious the recitations as set forth in amended independent Claim 11. Accordingly, reconsideration and withdrawal of this rejection are respectfully requested.

Independent Claim 18

The rejection of Claim 18 is respectfully traversed. It is respectfully submitted that the Fletcher and Johnston references, alone or in combination, fail to describe, teach, or suggest a processor for simultaneously executing: (1) a processor for simultaneously executing: (2) a protocol driver; (3) a network driver software interface; (4) a media access control unit; and (5) a rerouting driver, wherein during installation of the rerouting driver, (6) a first section of code in the network driver software interface is disabled (7) by overwriting code that is positioned before the first section of code with blocking code (8) so that complete operation of the network driver software interface is not stopped, and (9) wherein the first section of code is then patched (10) by

Serial No. 09/456,894

inserting a template jump (11) for forwarding call instructions from the network driver software interface to a template in the rerouting driver, (12) the template comprising new computer code for controlling communications, (13) the template jump and template allowing the rerouting driver to control communication between the one or more media access control units and the one or more protocol drivers;; (14) the network driver software interface to store a first binding defining a communication path between the protocol driver and the media access control unit, (15) the network driver software interface coupled to communicate packets with the media access control unit, (16) the network driver software interface being patched to communicate the packets with the rerouting driver; and (17) the rerouting driver being executed by the processor at the same time as the network driver software interface and (18) being coupled to communicate the packets with the protocol driver, as recited in amended independent Claim 18.

Similar to what is discussed above with respect to independent Claim 1, the Fletcher and Johnston references do not provide any teaching of (a) a processor for simultaneously executing: (b) a protocol driver; (c) a network driver software interface; (d) a media access control unit; and (e) a rerouting driver, wherein during installation of the rerouting driver, (f) a first section of code in the network driver software interface is disabled (g) by overwriting code that is positioned before the first section of code with blocking code (h) so that complete operation of the network driver software interface is not stopped, and (i) wherein the first section of code is then patched (j) by inserting a template jump (k) for forwarding call instructions from the network driver software interface to a template in the rerouting driver, (l) the template comprising new computer code for controlling communications, as recited in amended independent Claim 18.

In light of the differences between the claims and the Fletcher and Johnston references mentioned above, one of ordinary skill in the art recognizes that these references alone or in combination cannot anticipate or render obvious the recitations as set forth in amended independent Claim 18. Accordingly, reconsideration and withdrawal of this rejection are respectfully requested.

Serial No. 09/456,894

Independent Claim 23

The rejection of Claim 23 is respectfully traversed. It is respectfully submitted that the Fletcher and Johnston references, alone or in combination, fail to describe, teach, or suggest a rerouting driver comprising: (1) control code, for controlling the (2) rerouting driver; (3) binding code, for establishing at least one binding at the (4) network driver software interface so that the rerouting driver is bound to at least (5) one media access control unit (6) while the network driver software interface and the rerouting driver are executed at the same time; (7) patching code, (8) for inserting template jumps into at least a first section of code in the network driver software interface, (9) the template jumps providing jumps to templates in the rerouting driver so that information from at least one media access control unit destined for at least one protocol driver is rerouted to the rerouting driver (10) while the network driver software interface and the rerouting driver are executed at the same time; (11) at least one template, (12) for receiving information from at least one template jump in the network driver software interface and (13) comprising inserted code for controlling communications; (14) blocking code, (15) for preventing processing of the patching code that is positioned after the blocking code and (16) for allowing continuous operation of the network driver software interface; and (17) the inserted code, (18) for replacing the blocking code and (19) to evaluate rerouted information received by the template jumps, as recited in amended independent Claim 23.

Similar to what is discussed above with respect to independent Claim 1, the Fletcher and Johnston references do not provide any teaching of rerouting driver comprising: (a) control code, for controlling the (b) rerouting driver; (c) binding code, for establishing at least one binding at the (d) network driver software interface so that the rerouting driver is bound to at least (e) one media access control unit (f) while the network driver software interface and the rerouting driver are executed at the same time; (g) patching code, (h) for inserting template jumps into at least a first section of code in the network driver software interface, (i) the template jumps providing jumps to templates in the rerouting driver so that information from at least one media access control unit destined for at least one protocol driver is rerouted to the rerouting driver (j) while the network driver software interface and the rerouting driver are executed at the same time; and blocking code, (k) for preventing processing of the patching code that is positioned after the

Serial No. 09/456,894

blocking code and (l) for allowing continuous operation of the network driver software interface, as recited in amended independent Claim 23.

In light of the differences between the claims and the Fletcher and Johnston references mentioned above, one of ordinary skill in the art recognizes that these references alone or in combination cannot anticipate or render obvious the recitations as set forth in amended independent Claim 23. Accordingly, reconsideration and withdrawal of this rejection are respectfully requested.

Independent Claim 26

The rejection of Claim 26 is respectfully traversed. It is respectfully submitted that the Fletcher and Johnston references, alone or in combination, fail to describe, teach, or suggest the combination of (1) selecting a first section of code of the network driver software interface in (2) a central processing unit that is to be modified (3) while the network driver software interface is running; (4) writing the first section of code of the network driver software interface into the cache memory of the central processing unit (5) while the network driver software interface is running; (6) overwriting a portion of the first section of code in cache memory with blocking code comprising (7) code that causes a loop around serialization instruction in order (8) to create a first version of code while the network driver software interface is running; (9) writing the first version of code into shared memory (10) while the network driver software interface is running; (11) modifying the first version of code in the cache memory to create a second version of code, (12) wherein a portion of the code following the blocking code is overwritten with template jumps (13) for forwarding calls to a template (14) to effect a static patch of the network driver software interface when the network driver software interface is running in the shared memory, (15) the template comprising new computer code for controlling communications, (16) the template jump and template allowing the new computer code to control communication between one or more media access control units and one or more protocol drivers; (17) writing the second version of code into shared memory while the network driver software interface is running; (18) modifying the second version of code in the cache memory with code to create a third version of code, (19) wherein the blocking code is overwritten to remove the blocking code while the

Serial No. 09/456,894

network driver software interface is running; and (20) writing the third version of code into shared memory while the network driver software interface is running, as recited in amended independent Claim 26.

Similar to what is discussed above with respect to independent Claim 1, the Fletcher and Johnston references do not provide any teaching of the combination of (a) writing the first section of code of the network driver software interface into (b) the cache memory of the central processing unit (c) while the network driver software interface is running; (d) overwriting a portion of the first section of code in cache memory with blocking code comprising (e) code that causes a loop around serialization instruction in order (f) to create a first version of code while the network driver software interface is running; (g) writing the first version of code into shared memory (h) while the network driver software interface is running; (i) modifying the first version of code in the cache memory to create a second version of code, (j) wherein a portion of the code following the blocking code is overwritten with template jumps (k) for forwarding calls to a template (l) to effect a static patch of the network driver software interface when the network driver software interface is running in the shared memory, as recited in amended independent Claim 26.

In light of the differences between the claims and the Fletcher and Johnston references mentioned above, one of ordinary skill in the art recognizes that these references alone or in combination cannot anticipate or render obvious the recitations as set forth in amended independent Claim 26. Accordingly, reconsideration and withdrawal of this rejection are respectfully requested.

Independent Claim 28

The rejection of Claim 28 is respectfully traversed. It is respectfully submitted that the Fletcher and Johnston references, alone or in combination, fail to describe, teach, or suggest the combination of: (1) disabling access to a first section of code of a network driver software interface (2) while the network driver software interface is running by (3) overwriting code that is positioned before the first section of code with (4) blocking code, (5) the first section of code providing a communication path between a media access control unit and an application, (6) the

Serial No. 09/456,894

first section of code including a generic call; (7) overwriting the first section of code with a second section of code (8) while the network driver software interface is running the blocking code; (9) overwriting the blocking code with the second section of code and (10) without stopping complete operation of the network driver software interface; (11) executing the second section of code to cause execution flow to be rerouted to a third section of code in a rerouting driver, (12) the second section of code being no larger than the first section of code, (13) the third section of code, when executed and (14) while the network driver software interface is running the second section of code, completing the communication path and returning execution flow, (15) the third section of code including additional code not present in the first section of code that is now inserted into the communication path and (16) that controls communications between the media access control unit and an application, as recited in amended independent Claim 28.

Similar to what is discussed above with respect to independent Claim 1, the Fletcher and Johnston references do not provide any teaching of the combination of: (a) overwriting the first section of code with a second section of code (b) while the network driver software interface is running the blocking code; (c) overwriting the blocking code with the second section of code and (d) without stopping complete operation of the network driver software interface; (e) a third section of code including additional code not present in the first section of code that is now inserted into the communication path and (f) that controls communications between the media access control unit and an application, as recited in amended independent Claim 28.

In light of the differences between the claims and the Fletcher and Johnston references mentioned above, one of ordinary skill in the art recognizes that these references alone or in combination cannot anticipate or render obvious the recitations as set forth in amended independent Claim 28. Accordingly, reconsideration and withdrawal of this rejection are respectfully requested.

Dependent Claims 2, 4-10, 12, 14-17, 19-22, 24-25, and 29

The Applicant respectfully submits that the above-identified dependent claims are allowable because the independent claims from which they depend are patentable over the cited

Serial No. 09/456,894

references. The Applicant also respectfully submits that the recitations of these dependent claims are of patentable significance.


In view of the foregoing, the Applicant respectfully requests that the Examiner withdraw the pending rejections of dependent Claims 2, 4-10, 12, 14-17, 19-22, 24-25, and 29.

CONCLUSION

The foregoing is submitted as a full and complete response to the Office Action mailed on December 16, 2005. The Applicant and the undersigned thank Examiner El-Chanti for consideration of these remarks. The Applicant has amended the claims and has submitted remarks to traverse rejections of Claims 1-2, 4-12, 14-26 and 28-29. The Applicant respectfully submits that the present application is in condition for allowance. Such action is hereby courteously solicited.

If the Examiner believes that there are any issues that can be resolved by a telephone conference, or that there are any formalities that can be corrected by an Examiner's amendment, please contact the undersigned in the Atlanta Metropolitan area (404) 572-2884.

Respectfully submitted,


Steven P. Wigmore
Reg. No. 40,447

June 15, 2006

King & Spalding, LLP
45th Floor
191 Peachtree Street, N.E.
Atlanta, Georgia 30303
404.572.4600
K&S Docket: 05456.105030